

Multi-resolution Texture Visualization of Vector Field Based on Fuzzy Feature Measurement

Xu Huaxun, Ma Qianli, Li Sikun, Zeng Liang, Cai Xun

Abstract—Effectively visualizing the feature structures is a challenging work due to the occlusion and clutter constraints in the 3D flow visualization, especially for the texture-based approaches. This paper proposes a multi-resolution texture-based method for visualizing 3D flows based on a fuzzy flow feature extraction method. Existing algorithms always concretely use topology separatrix to partition a flow field into regions of different behaviors. However, flow feature region usually doesn't have distinct discrepancy and couldn't be clearly defined. Therefore, guided by the fuzzy theory, we present a fuzzy feature extraction approach which treats the flow features as fuzzy sets and establishes some fuzzy rules to extract them. In the rendering stage, we apply multi-resolution technology to the sparse texture-based method, which are widely used for the advantage of continuous and global representation of flow movements. Based on the fuzzy membership field for flow features, we present a multi-resolution algorithm to prefilter the input noise. The resulted sparse textures could distribute reasonably according to feature structures. Experiments on synthetic and real engineering dataset demonstrate that our approach can depress the occlusion and clutter effectively.

Index Terms—vector field, sparse texture, flow feature, fuzzy theory.

I. INTRODUCTION

Texture-based techniques become an effective choice for visualizing vector field. The resulting texture is a powerful way to convey essential patterns of the vector field, while avoiding the tedious task of seeding individual streamlines to capture all the structures of interest. Texture-based approaches can be loosely classified into two categories: line integral convolution (LIC) [1] and texture advection [2]. The LIC method adopts a noise texture as input, and then the noise is convolved in the direction of the flow using filter kernels. In texture advection, the basic idea is advecting the texel in the direction of the flow to generate the final visual pattern.

In two-dimensional vector fields, the texture-based methods are capable of offering a clear representation since it can create a continuous visual representation to illustrate the global behavior of the flow. For three-dimensional vector fields, however, it can be challenging to find a good visual representation of the flow due to the loss of information when the three-dimensional data is projected onto a two-dimensional image plane. And the occlusion and cluttering problem encountered when extending the previous approaches to three-dimension make the clear representation

of vector fields impossible. Especially for dense texture-based techniques, representations of 3D flows make the inner feature structures become more occlusive.

Perception can be improved by highlighting important features in the flow fields. Feature-based approach is developed to extract some special structures, or some specific patterns of interest, from the flow datasets. There are various features, such as vortices, shock waves, or separatrices, in the flow field. Each of them has specific physical properties, which can be used to extract the desired feature. However, the explicit or clear feature extraction is a harsh and hard term, since there always is no distinct discrepancy among different feature structures in flow fields. Furthermore, some important structures such as vortex or turbulence are so complicatedly mingled that they can't explicitly expressed by concreated feature primitives. Worsely, these features don't have accurate definitions at all up to now.

This paper presents a novel texture-based method for visualizing 3D flow fields which utilize feature-based and texture-based approaches. To obtain the various feature structures in the flow field, we present a feature extracted algorithm based on fuzzy theory. The main advantage of this algorithm is it's capability of extracting different flow features in a uniform way. As for rendering stage, we proposed a multi-resolution texture method based on the feature structures extracted to relieve the occlusion and cluttering phenomena, some of our experiment results are showed in Figure 1. The key contributions of this paper are: (1) a flexible fuzzy feature region extraction method. By expressing feature regions as a fuzzy membership field, we could define more kinds of feature structures with arbitrary topology shape in a uniform way. To a new feature structure, it is easy to add new fuzzy measure rules into the current algorithm without changing its basic structure. (2) a multi-resolution rendering technique. Based on the fuzzy feature measurement extracted, we generate a multi-resolution input noise for LIC convolution to obtain a dense representation of feature structures in the flow, while rendering the background flow movement in a sparse manner.

The remainder of this paper is organized as follows. Related work is discussed in section 2. A detailed explanation of our feature extraction method is provided in section 3. In section 4, we present the multi-resolution noise generation process. Next we will show the experiment results of our method in section 5. Finally, we conclude by pointing out interesting avenues of future work in section 6.

II. RELATED WORK

There exist a large number of different 3D flow feature visualization techniques, which can be distinguished according to their properties w.r.t a number of categories. We would briefly review most related work and the relevant techniques

Manuscript received April XX, 20XX; revised June XX, 20XX. This work was financially supported by the National Grand Fundamental Research 973 Program of China under Grant No.2009CB723803, and by Natural Science Foundation of China (No.60873120).

Xu Huaxun, Ma Qianli, Li Sikun, Zeng Liang, Cai Xun are with the School of College of Computer Science, National University of Defense Technology, Changsha city, Hunan province 410073 China e-mail: xhxnudt@gmail.com.

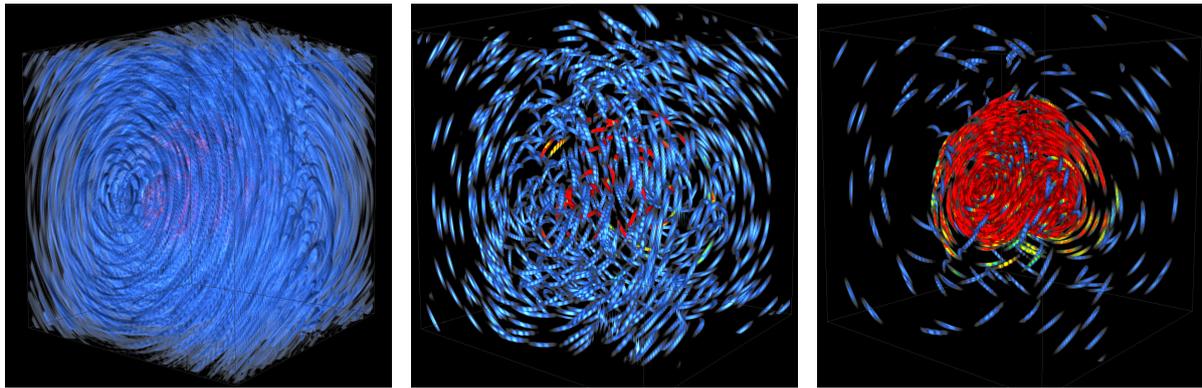


Fig. 1. The integrated Lorenz attractor. Left: The important attractor region are occluded with the context when adopting dense texture-based method. Middle: The sparse texture method can resolve the occlusion problem, yet some information about the two spiral attractor structures is also lost. Right: The occlusion and clutter problems are solved by our multi-resolution approach completely, the Lorenz attractor is effectively visualized in an efficient way.

employed in our system, since there is a large literature on this subject. For more detailed discussion on these topics, we refer to related surveys [3]–[5].

Feature-based approach is developed to extract some special structures in flow fields by extracting physically meaningful patterns from the data sets. A popular method used is topological analysis, which first introduced to the 3D visualization community by Helman and Hesselink [6]. Recently, [7] introduced an approach to visualize higher order critical points of 3D vector fields. [8] examined a high-dimensional, massive flow data set around an airfoil using a topology-based vortex analysis. [9] extracted singularities and separatrices to decompose the flow field into topological regions. To some special feature structures, such as vortex, a local criterion proposed by Jeong and Hussain [10]. [11] introduced a system which can analyze the vortex structures contained in the turbulent flows by means of a novel vortex core line detection method. [12] extend the method by a local statistical complexity approach to automatically detect distinctive structures in time-dependent multi-fields. [13] proposed a novel vortex core line extraction method based on the vortex region criterion in order to improve the detection of vortex features for 3D flow visualization. [14] introduced a new criterion to characterize hierarchical two-dimensional vortex regions induced by swirling motion, they defined vortex regions as closed loops that intersect the flow field at a constant angle and presented a parameter free algorithm for the identification of these regions.

Texture-based methods also are an important way to visualize vector field. The advantage of this kind of approaches is that they can represent global behavior of the flow in continuous visual representation. A more detailed discussion on these topics can refer to related surveys [15]. Early techniques include spot noise [16] and Line Integral Convolution (LIC) technique [1]. Some related approaches adopt texture advection [2] technology, including Lagrangian-Eulerian Advection (LEA) [17] and Image-Based Flow Visualization (IBFV) [18] approaches.

Recent advances in texture-based visualization extend the above-mentioned techniques into 3D vector fields. Shen et al. presented an interactive texture-based algorithm for visualizing three-dimensional vector fields, which allowing the user to compute three dimensional flow textures interactively

and to modify the appearance of the visualization on the fly [19]. Bachthaler et al. introduced an orthogonal vector field visualization method based on 2D manifolds [20]. Weiskopf et al. proposed an interactive technique for the dense texture-based visualization of unsteady 3D flow, taking into account issues of computational efficiency and visual perception. A primary challenge for dense texture-based methods is occlusion and cluttering problem, due to the dense rendering objects will be occluded by others. Some previous works are capable of producing a global but sparse representation of steady 3D vector fields to reduce the occlusion phenomena [19], [21]. Falk et al. presented a largely output-sensitive visualization method for 3D line integral convolution, which also adopting sparse streaks to avoid the same perceptual issues [24].

In contrast to previous concrete illustrations of flow feature, we utilize fuzzy theory to express the holistic/rough overview over the dataset and our method do not require a clear classification into interesting or not. Relevant regions can be identified subsequently as a subset of the rough partition with different flow behavior. Some features in 3D flows are very complex and our knowledge about them is incomplete, so their regions can't be defined clearly. We believe that fuzzy theory can overcome this shortcoming for its capability of describing objects without complete knowledge. Based on the feature structures extracted, multi-resolution ideas can be adopted to avoid the occlusion and cluttering for 3D texture-based approaches.

III. FUZZY MEASUREMENT OF 3D FLOW FEATURE

In this section, we would introduce the mathematical description for 3D flow features guided by fuzzy theory. Following the brief review of topology analysis of 3D flows (Section 3.1), the related definitions and rules are addressed in Sections 3.2, and how to calculate the fuzzy membership degree will be discussed in Sections 3.3.

A. Representation of 3D Flow Based on Fuzzy Theory

After the classification of critical points in 3D flows, we re-define the related concepts, generally used in feature visualization, guided by the fuzzy theory.

Definition1 Flow Field: The flow field can be regarded as a triple $U(\Phi, \Upsilon, \Pi)$, including basic properties Φ , derived properties Υ and associated properties Π .

The basic properties including position and velocity, which is the primary attributes of the flow fields. The derived properties, such as gradients and vortices, are derived from the basic properties. The associated properties are temperature, pressure and some other physical properties which are closely related with the basic properties. Based on the definition above, we introduce the flow feature and other related concepts here from the perspective of fuzzy theory.

Definition2 Flow Feature: The flow feature in the flow field $U(\Phi, \Upsilon, \Pi)$ is a fuzzy subset F^i , which has one special attribute i in the 3D flow X , it can be regarded as a ordered pair:

$$F^i = \{x, \mu_{F^i}^i(U(x))\}, x \in X;$$

$$\mu_{F^i}^i(U(x)) = \mu_{F^i}^i(\Phi(x), \Upsilon(x), \Pi(x)) \in [0, 1]$$

The mapping $\mu_{F^i}^i(x)$ is membership function, which mapping a $\forall x \in X$ to a certain value in the range $[0, 1]$, named as membership degree of x for feature subset F^i . When one position have greater value x , it means that the possibility of this point subjecting to a feature region is larger.

Definition3 Feature Region: If F^i is a flow feature of property i , then the feature region of F^i is a crisp set containing all points meeting $\mu_{F^i}^i(x) > 0.5$ in the universe of domain X , which corresponds to the support set of F^i and denoted by $SuppF^i$ or F_0^{i+} , that is:

$$F_1^i = KerF^i = \{x | \mu_{F^i}^i(x) = 1\}$$

The advantage of definition by using fuzzy subset is able to qualitatively describe flow feature. For instance, though we don't understand vortex feature clearly, but we know that if a position has greater λ_2 value, then it has greater possibility fell into the vortex region.

B. Fuzzy Rules of Feature Region Measurement

After the redefinitions of flow feature and some related concepts based on fuzzy theory, the question remaining is how to judge a point in flow field is whether in a special feature region or not. Based on the definitions introduced in section 3.2, we build relevant fuzzy rules from three levels, including basic properties, derived properties and associated properties, to divide the whole field into feature regions and the context.

Φ Rule: If C_0^i represents a critical point set, and $C_0^i \subseteq X$, we then have

$$\forall x \in C_0^i \mapsto \mu_{F^i}^i(x) = 1$$

By Φ rule, we get the cores of flow features including sinks, sources and saddles, as described in section 3.1. Certainly, there are other methods to get cores for some special features, such as minimum λ_2 approach for vortex. But we want to represent general topology feature in this paper, so we just introduced this rule.

In a flow field, the physical property such as heat will propagate along streamlines. So the feature region in flow field should reflect this directional property. To describe this property, we introduce the concept of streamline distance firstly.

Definition6 Streamline Distance: For any two points in the vector field, the streamline distance is defined as following:

$$L = \begin{cases} \int_A^B f(x)dx & \text{iff existing a streamline } f \text{ cross } A, B \\ \infty & \text{otherwise} \end{cases}$$

The streamline distance represents the space between two positions along the streamline. Based on this concept, we build the first derived property rule:

Υ .1Rule: For any point x, y in the vector field, if x is closer to one critical point c than y in streamline distance, then x has greater degree of membership of c :

$$\forall x, y \in U, c \in C^i, L(x, c) < L(y, c) \mapsto \mu_{F^i}^i(U(x)) > \mu_{F^i}^i(U(y))$$

To sources and sinks, the above rule is able to reflect their feature region effectively, as shown in figure 3. However, saddles are composed by one pair of hyperbolic surfaces. If we only use streamline distance, it should be unsatisfactory for extracting the feature region in the hyperbolic sector and the elliptic sector, since there is no streamline which come from the critical point in these parts. At the same time, there are some structures like close orbits, their streamline distances from any critical point are infinite, so we can't compute its membership degree by streamline distance. Therefore, we introduce the following rule based on Euler distance to solve this problem:

Υ .2Rule: For any point x in vector field, if its Euler distance from one critical point c is smaller, then its membership degree subjecting to c is greater:

$$\forall x, y \in X, c \in C^i, D(x, c) < D(y, c) \mapsto \mu_{F^i}^i(U(x)) > \mu_{F^i}^i(U(y))$$

As for vortex, which can seem as special sink type in some aspects, we establish a measurement rule based on λ_2 method [12], [13], [25].

Υ .3Rule: For any point x with less negative λ_2 value than point y in the vector field, then its membership degree subjecting to one vortex c is greater:

$$\forall x, y \in X, \lambda_2(x) < \lambda_2(y) \mapsto \mu_{F^i}^i(U(x)) > \mu_{F^i}^i(U(y))$$

In addition, there exit other attribute fields such as temperature, pressure and etc. Therefore, we build the following rule between the cyclone and the low pressure center for typhoon dataset.

Π .1Rule: If we have corresponding pressure field, then the point x which have a smaller pressure value than y will have a greater membership degree to one sink point:

$$\forall x, y \in X, P(x) < P(y) \mapsto \mu_{F^i}^i(U(x)) > \mu_{F^i}^i(U(y))$$

There may be some other rules for depicting flow features. The advantage of our method is its flexibility for adding new rule without changing its basic framework. To our tested datasets, these rules are enough for extracting the user-interested feature regions.

C. Fuzzy membership calculation

Based on the established rules, we propose a fuzzy k-means-like method to cluster feature regions. Firstly, we build a characteristic vector for each sample point in the 3D flow fields according to the relevant rules.

Assuming there are c points in the flow field \mathbf{X} , and the current sample point is x_k . The essence of the $\Upsilon.1$ rule is to judge whether the current sample point x_k and the critical points are in the same streamline, so we build the following c -dimensional character vector for the $\Upsilon.1$ rule:

$$V_{\Upsilon.1}(x_k) = (r_{k1}, r_{k2}, \dots, r_{kc}),$$

where $r_{kj}(1 \leq j \leq c) = \begin{cases} 0 & \text{iff } L(x_k, c_j) = \infty \\ 1 & \text{otherwise} \end{cases}$ Where c_j is the j -th critical point. For example, if x_k is at the streamline which starts from c_j and ends at c_l , then we have $V_{\Upsilon.1}(x_k) = (0, 0, \dots, r_{kj} = 1, 0, \dots, r_{kl} = 1, \dots, 0)$.

The $\Upsilon.2$ rule represents the Euler distance between the sample point x_k and the critical points, so we use the sample point coordinate and build the character vector $V_{\Upsilon.2}(x_k) = (x_k, y_k, z_k)$.

To the $\Upsilon.3$ and $\Pi.1$ rules, we also build the following character vectors $V_{\Upsilon.3}(x_k) = \lambda_2(x_k)$ and $V_{\Pi.1}(x_k) = p(x_k)$.

Consequently, we get a $c+5$ dimensional character vector for every sample point:

$$V(x_k) = (V_{\Upsilon.1}(x_k), V_{\Upsilon.2}(x_k), V_{\Upsilon.3}(x_k), V_{\Pi.1}(x_k))$$

After the computation of character vectors, we normalize the vectors to convert them into the range $[0, 1]$. Because critical points are the centers of feature regions in the flow field \mathbf{X} , we regard them as the standard samples.

Finally, the membership degree of is computed by the following formula (13). Similar to [22], this approach can be proved to be the optimal in the minimum-square-sum rule.

$$\mu_{ik} = \begin{cases} \sum_{j=1}^{c+1} \frac{d_{ik}}{d_{jk}} & \text{if } I_k = \phi \\ 1 - \frac{L(x_k, c_i)}{L_{len}(x_k)} & \text{if } I_k \neq \phi, i \in I_k \text{ and } i \leq c \\ 0 & \text{if } I_k \neq \phi, i \in \bar{I}_k \text{ and } i \leq c \\ 1 - \sum_{i=1}^c \mu_{ik} & \text{otherwise} \end{cases} \quad (1)$$

Where $I_k = \{i | 1 \leq i \leq c, d_{ik} = 0\}$, $\bar{I}_k = \{i | 1 \leq i \leq c\} - I_k$, and $d_{ik} = \|V(x_k) - V(c_i)\| = \left(\sum_{j=0}^{c+5} (V_j(x_k) - V'_j(c_i))^2 \right)^{1/2}$ is the deviation between the current point and the related standard sample c_i , and $L(x_k, c_i)$ is the streamline distance between x_k and c_i , $L_{len}(x_k)$ denotes the length of the streamline across the x_k in the flow field \mathbf{X} .

IV. MULTI-RESOLUTION LINE INTEGRAL CONVOLUTION

Next we will introduce our noise generation method firstly, and then present the multi-resolution controlled approach proposed by this paper. The sparse flow textures are created by an output-sensitive LIC algorithm, which accelerated the line integral convolution process by skip the part where don't have contribution on the final rendering result.

A. Noise Generation

The original input noise for LIC algorithm is white noise, which generates a pseudo-random number sequence. A drawback of this approach is the sparse noise texels generated are not even enough to cover the vector field well-proportioned. This is why some papers adopted low discrepancy sequence

[23], which tend to provide a more well-distributed set of samples than pseudo-random method.

Low discrepancy sequence is a point sets which can acquire the minimum order of magnitude of the estimation error in quasi-Monte Carlo methods. The most simple low discrepancy sequences are Hammersley sequence and the Halton point set. But the Hammersley sequence is finite, so the sample number N is restrained to a fixed number of points. So we use the closely related Halton sequence here for seed generation, it has the nice property that new seeds can be adaptively generated [23]. The common low discrepancy sequence can be generated by radical inverse function, which is showed as following:

$$\Phi_b(i) := \sum_{j=0}^{\infty} a_j(i) b^{-j-1} \Leftrightarrow i = \sum_{j=0}^{\infty} a_j(i) b^j \quad (2)$$

Here b is the base number, and $\Phi_b(i)$ represents the expression of i in base b , and the final i -th sequence number with b can be calculated as the right part of label \Leftrightarrow . Based on the inverse function, the Halton sequence are constructed by $x_i = (\Phi_{b_1}(i), \dots, \Phi_{b_n}(i))$ where b_1, \dots, b_n are the first prime numbers.

After the construction of the Halton point sets, we use Gaussian filter to smear out the sparse noise ball. This filter has an advantage of an exponential fall-off property, whether in spatial space or frequency space. Depending on the filter kernel width, the final LIC sparse streamlines of different thickness can be achieved.

B. Multi-resolution control

The resulting sparse noise generated by previous methods [24] distributed evenly in the vector field. One shortcoming of this approach is the accurate information on feature structures or the region of user interested could be neglected in the final rendering results. The information of some flow region interested may be lost when the resulted noise is too sparse, while dense noise may arouse the occlusion problem. So it is reasonable to visualize different regions with different degrees of accuracy on a single image.

The sparse multi-resolution noise generation process is given in the algorithm 1, where M denotes the feature measure value obtained in the section 3, and N is the output sparse noise. Because we can treat the flow feature as a fuzzy sets, so the measure value is a number in the range $[0,1]$. It can be seen as the probability of one voxel in vector filed subjected to the feature region. So the output sparse noise of algorithm 2 guarantees the feature regions in the flow have a more detailed representation than the unimportant areas.

C. Output-Sensitive Line Integral Convolution

In this paper, the texture-based visualizing strategy is based on the LIC approach, and we adopted sparse texture method to avoid the perception difficulties. The LIC evaluates the final texture value at position as following:

$$\rho(\mathbf{r}) = \int_{s_0-L_s}^{s_0+L_e} k(s-s_0) N(\sigma(s)) ds$$

where $N(\sigma(s))$ denotes the noise function N value at the sample point $\sigma(s)$ and the k are the filter kernel functions. From the above formula, it can see that the original input

Algorithm 1 Multi-resolution Noise Generation

Input : a, M

Output : N

```
1: while  $n < N$  do
2:   Generate a random position  $P$  in vector field.
3:   Obtain the measure membership value  $M[P]$  obtained in section 3.
4:   Generate a random number  $r$  in  $[0, 1]$ .
5:   if  $r * M[P] > a$  then
6:     Filtered by Gaussian kernel function, with  $P$  as center position.  $n++$ .
7:   else
8:     Reject the position  $P$  as seed points;
9:   end if
10: end while
```

noise is instrumental in determining the representation of the flow visualization.

Previous LIC approaches separated the computing process of 3D noise $\rho(r)$ from the step of volume rendering integral. This suffers from a main disadvantage is an unnecessary evaluation of the LIC integral for points that are invisible due to occlusion or feature extraction. To solve this problem, we adopted the output sensitive LIC approach mentioned in [24], which use an on-the-fly computation of the LIC integral during volume rendering. The basic idea is that the rendering cost should be proportional to the complexity of the contributions to the information on the image plane and independent of the data set size. So it adopted some culling acceleration techniques such as early-ray termination, which are extremely beneficial to improve the render performance since the LIC computation is not executed for noncontributing samples.

Because the distribution of input spares noise in our method is proportional to the importance of flow feature, so there are lots of unimportant regions which don't influence the final rendering result can be culled early. This can improve the time performance of LIC computation, while maintaining a high quality for the flow visualization.

V. RESULTS AND DISCUSSION

We tested our approach on numerical and synthetical simulation datasets. For the numerical case, we examined a spacecraft dataset, coming from real engineering project. For synthetical flow data, we examined the classical Lorenz attractor $u = (10(y - x), 28x - y - xz, xy - (8/3)z)$. The experiments we have implemented make full use of GPU acceleration technologies to improve the rendering performance.

As for the spacecraft dataset, our method also can extract and show vortex feature regions effectively. As showed in Figure 2(a), the texture visualizing result based on the input noise without prifiltered by the algorithm 1, and the vortex structures are occluded by the around sparse LIC streaks. This problem is relieved greatly in the Figure 2(b), which adopts the multi-resolution approach presented in this paper. The vortex regions are emphasized by dense texture visualization, while the background flows are showed in a sparse manner.

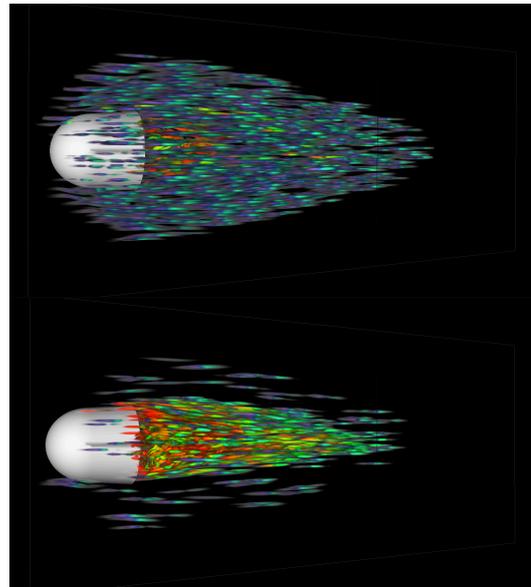


Fig. 2. Our method can depict feature regions more clearly in the overall flows with multi-resolution method(b). While previous method [24], adopting evenly distributed noise, can't represent the vortex structure very well since the occlusion problem(a).

We also used two cool/warm tone based illumination methods [26] in our experiments. As shown in Figure 3, the upper row uses noise droplets distribute evenly in the vector field, while the lower row adopted our multi-resolution prefiltering algorithm and the noise droplets density is proportional to feature structure membership value. Figure 3(a) used a scumbled cool/warm illumination model, and it can reflect the flow direction very well in an overall manner. However, the combination this method with the input noise generated by algorithm 2 loses its advantage of representing the concrete flow direction. Compared with this method and traditional gradient-based method (see Figure 3(c)), the exquisite model is able to reflect the direction of flow clearly in any situation, as showed in the Figure 3(b) and (d). Furthermore, the occlusion phenomena can be solved very well with our multi-resolution approach, as showed in Figure 3(c) and (d).

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an interactive particle-based method for visualizing vector fields, capable of generating a global but sparse representation of the flow. This method regards the flow feature as fuzzy subset and establishes some rules to calculate the flow feature membership. Once the membership field calculation is completed, the output data are sent to graphic hardware as a texture to identify the importance of each point in the field.

In the rendering stage, this method adopts multi-resolution texture technologies based on the feature membership field. The sparse texture streaks distribute proportionally to the importance degree. In this way, the feature structures can be illuminated by a dense texture streaks while context region in a sparse density. We also used two illumination model based on cool/warm lighting technology. Our illumination methods can overcome the defect of texture-based visualization, which unable to reflect the concrete flow direction.

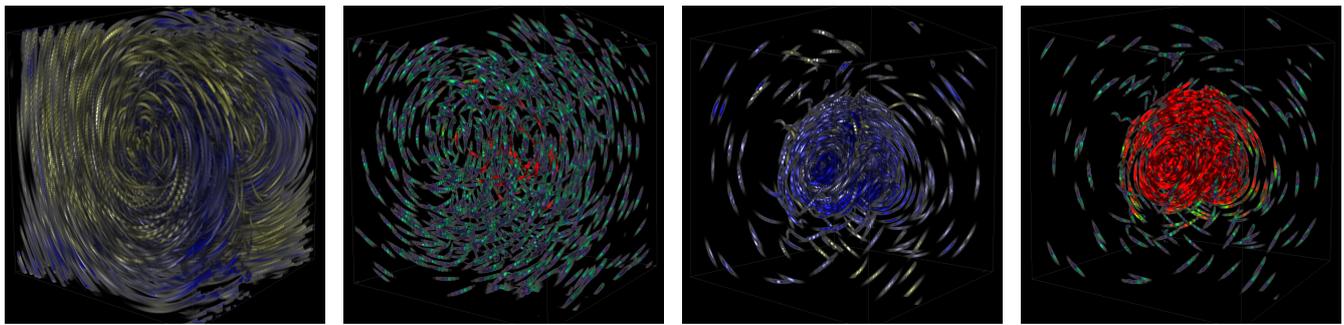


Fig. 3. From left to right, rendering lorinz dataset with: (a) evenly distributed input noises and scumbled cool/warm illumination model, (b) evenly distributed input noises and exquisite cool/warm illumination model, (c) multi-resolutional distributed input noises and the gradient-based illumination model, (d) multi-resolutional distributed input noises and the exquisite model.

Our method have some weaknesses to improve in the future. For instance, it is time-consuming when the flow structures are complicated. We will continue to simplify the calculation of membership value and explore further benefits of having multiple graphics cards. Another problem to be solved is the reasonable textures density control in algorithm 1. As showed in the Figure 2, the density of sparse textures are too large so that the occlusion phenomenon is still serious. And for some complex features, such as turbulence, we don't have taken into consideration now. A possible extension is to build some new measure rules to resolve this problem. Another possible future work is exploring multi-dimensional transfer function and extending our approaches to unstructured and irregularly grid-structured data.

REFERENCES

- [1] B. Carbal , L.C. Leedom, "Imaging vector fields using line integral convolution," In ACM SIGGRAPH Conference 1993, pp. 263-270.
- [2] N. Max and B. Becker, "Flow Visualization Using Moving Textures," In Proceeding of ICASW/LARC Symp. Visualizing Time-Varying Data 1995, pp. 77-87.
- [3] F.H. Post, B. Vrolijk, H. Hauser, R.S. Laramée, H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," Computer Graphics Forum, 2003, 22(4), 775-792.
- [4] C. Garth, X. Tricoche, "Topology and feature based flow visualizaion: methods and applications," In Proceedings of SIAM Conference on Geometric Design and Computing 2005.
- [5] S. Tobias, J. Heike, W. Thomas, S. Gerik, "The State of the Art in Flow Visualization: Partition-Based Techniques," In Proceedings of Simulation and Visualization, 2008, pp. 75-92.
- [6] J.L. Helman, L. Hesselink, "Visualizing Vector Field Topology in Fluid Flows," IEEE Computer Graphics and Applications. 1991, 11(3), 36-46.
- [7] T. Weinkauff, H. Theisel, K. Shi, et al., "Structure accentuating Dense Flow Visualization," In Proceedings of IEEE Visualization 2005, pp. 559-566.
- [8] T. Weinkauff, J. Sahner, B. Gunther, H. Theisel, H.C. Hege, F. Thiele, "Feature-based Analysis of a Multi-Parameter Flow Simulation," In Proceedings of SimVis 2008, pp. 237-252.
- [9] W. Keqin, L. Zhanping, Z. Song, J. Robert , II. Moorhead, "Topology-Aware Evenly-Spaced Streamline Placement," IEEE Transactions on Visualization and Computer Graphics(03), IEEE computer Society Digital Library(2009).
- [10] J. Jeong, F. Hussain, "On the identification of a vortex," Journal of Fluid Mechanics, 1995, 28(5), 69-94.
- [11] S. Stegmaier ,U. Rist , T.Ertl, "Opening the can of worms: An exploration tool for vortical flows," In Proceedings of IEEE Visualization 2005, pp. 463-470.
- [12] H. Janicke, A. Wiebel, G. Scheuermann , W.Kollmann, "Multifield Visualization Using Local Statistical Complexity," In Proceedings of IEEE Visualization 2007 ,pp. 1384-1391.
- [13] T. Schafhitzel, J.E. Vollrat, J.P. Gois, D. Weiskoph, A. Castelo, T. Ertl, "Topology- Preserving l2-based Vortex Core Line Detection for Flow Visualization," Computer Graphics Forum, 2008, 27(3), 1024-1030.
- [14] P. Christoph, K. Jens, P. Steffen, H. Hanschristian , "Hierarchical Vortex Regions in Swirling Flow," Computer Graphics Forum, 2009, 28(1), 863-870.
- [15] R.S. Laramée, H. Hauser, H. Doleisch, B. Vrolijk, F.H.Post, and D.Weiskopf, "The State of the Art in Flow Visualization: Dense and Texture-Based Techniques," Computer Graphics Forum, 2004, 23(2), 143-161.
- [16] J.J. van Wijk, "Spot Noise-Texture Synthesis for Data Visualization," Computer Graphics, 1991, 25, 309-318.
- [17] B. Jobard, G. Erlebacher, and M.Y.Hussaini, "Lagrangian- Eulerian Advection of Noise and Dye Textures for Unsteady Flow Visualization," IEEE Trans. Visualization and Computer Graphics, 2002, 8(3), 211-222.
- [18] J.J. van Wijk, "Image Based Flow Visualization," ACM Trans.Graphics, 2002, 21(3), 745-754.
- [19] Han-Wei Shen, Guo-Shi Li, Udepta D. Bordoloi, "Interactive Visualization of Three-Dimensional Vector Fields with Flexible Appearance Control," IEEE Transactions on Visualization and Computer Graphics, 2004, 10(4), 434-445.
- [20] S. Bachthaler and D. Weiskopf, "Animation of Orthogonal Texture-Based Vector Field Visualization," In Proceedings of Eurovis 2007, pp. 219-226.
- [21] S. Guthe, S. Gumhold, and W. Strager, "Interactive Visualization of Volumetric Vector Fields Using Texture Based Particles," In Proceeding of Central Europe Computer Graphics and Visualization 2002.
- [22] HB. Gao, "Fuzzy Cluster Analysis and its application," Xi'an, Xidian University press, 2004.
- [23] H. Niederreiter, "Random Number Generation and Quasi-Monte Carlo Methods," Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.
- [24] M. Falk, D. Weiskopf, "Output-Sensitive 3D Line Integral Convolution," IEEE Transactions on visualization and computer graphics, 2008, 14(4), 820-834.
- [25] S. Stegmaier, T. Ertl, "A Graphics Hardware-based Vortex Detection and Visualization System," In Proceedings of IEEE Visualization 2005, pp. 195-202.
- [26] Kai Yue, Hua-Xun.Xu, Xun.Cai, Si-Kun.Li, "A LIC Method Based on Cool/Warm Tone," In Proceedings of International Conference on Audio, Language and Image Processing (ICALIP), 2010.